Constructing Inter Domain Packet Filter for Controlling IP Spoofing

Madhuri Ghorpade

Student, Dept. of Computer Science, California State University Sacramento, California, USA

Abstract - *The Distributed Denial-of-Service (DDoS) attack is always been a serious threat to the legitimate use of the Internet. All impediment mechanisms employed have been prevented by the capacity of attackers to spoof the source address of the IP packet. By applying IP Spoofing, attackers can evade detection and put a substantial burden on the destination network for policing attack packets. An Inter Domain Packet Filter (IDPF) architecture is proposed to reduce the extent of IP spoofing on the internet. IDPF does not require global routing information. It is constructed based on the information implicit in Border Gateway Protocol (BGP) route update, and then they are deployed in network border routers. The conditions under which the IDPF discards packets with spoofed source address and allows packets with valid source addresses is established. Extensive simulation studies states that even with partial deployment on the internet, IDPFs can limit the spoofing capability of attackers. Further, they can help localize the source of an attack packet to a small number of candidate networks.*

Key Words: **IP Spoofing, IDPF, DDoS, BGP, legitimate use.**

1. INTRODUCTION

Distributed Denial of Service (DDoS) attacks pose an increasingly grave threat to the Internet, as evidenced by recent DDoS attacks mounted on both popular Internet sites and the Internet infrastructure [1]. IP spoofing is one of the factors that complicates the mechanisms of policing such attacks, the act of forging the source addresses in IP packets. Attackers hide its actual identity and location by masquerading as a different host, rendering source-based packet filtering less effective. It has been shown that a large part of the Internet is vulnerable to IP spoofing [3].

It has been observed that attackers can insert arbitrary source addresses into IP packets, but they cannot, however, control the actual paths through which the packets travel to the destination. Based on this observation, we proposed the route-based packet filters as a way to control IP spoofing. The basic idea in this scheme is that, assuming single-path routing, there is exactly one single path p(s, d) between source node s and destination node d. Hence, any packets with source address s and destination address d that appear in a router not in p(s, d) should be discarded. However, constructing a specific route-based packet filter in a node requires the knowledge of global routing decisions made by all the other nodes in the network, which is hard to reconcile on the current BGP-based Internet routing infrastructure [5].

Inspired by the idea of route-based packet filters, Inter Domain Packet Filter (IDPF) architecture has been proposed. The IDPF architecture takes advantage of the fact that while network connectivity may imply a large number of potential paths between source and destination domains, commercial relationships between ASes(Autonomous System) act to restrict to a much smaller set, the number of feasible paths that can be used to carry traffic from the source to the destination[7].

It is investigated how other AS relationship and routing information may help further improve the performance of IDPFs in our future work. The system shows that locally exchanged routing information between neighbors, i.e., BGP route updates, is sufficient to identify feasible paths and construct IDPFs, assuming all ASes on the Internet employ a set of routing policies that are commonly used today [15, 10]. Like route-based packet filters [4], the proposed IDPFs cannot stop all spoofed packets. However, when spoofed packets are not filtered out, IDPFs can help localize the origin of attack packets to a small set of ASes, which can significantly improve the IP traceback situation [5].

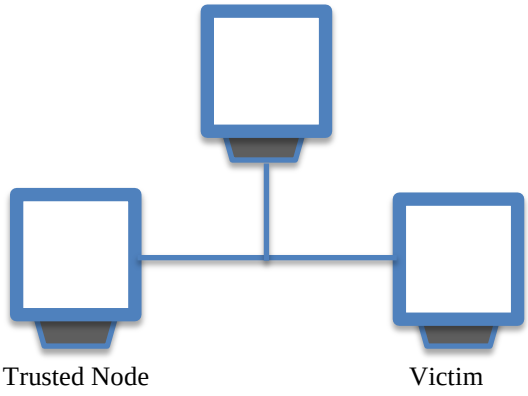

Fig -1: Intruder in communication

Internet Protocol (IP) is a basic protocol for transferring data over the Internet network. IP spoofing is the method of creating an IP packet using a fake IP address that is duplicating a legal and legitimate IP address with the intention of attacking a network in order to gain illegitimate access. The attack is based on the fact that origination address is generally ignored by routers while handling internet connection between distant computers. Routers finds the best route by examining the destination address.

2. RELATED WORK

The idea of IDPF is based on the study of the relationship between network topology and the effectiveness of route-based packet filtering. Unicast reverse path forwarding (uRPF) [9] requires that a packet is forwarded only when the interface that the packet arrives on is exactly the same used by the router to reach the source IP of the packet. Packet is dropped if the interface does not match. While simple, the scheme is limited given that Internet routing is inherently asymmetric, i.e., the forward and reverse paths between a pair of hosts are often quite different. Hence, the loose mode is less effective in detecting spoofed packets. In Hop-Count Filtering (HCF) [10], each end system maintains a mapping between IP address aggregates and valid hop counts from the origin to the end system. Packets that arrive with a different hop count are suspicious and are therefore discarded or marked for further processing. In Path Identification [11], each packet along a path is marked by a unique Path identifier (Pi) of the path. Victim nodes can filter packets based on Pi carried in the packet header. StackPi [12] improved the incremental deployment property of Pi by proposing two new packet marking schemes. In the Packet Passport System [13], a packet originated from a participating domain carries a passport that is computed based on secret keys shared by the source domain and the transit domains from source to destination. Packets carrying an invalid passport are discarded by the transit domains.

In the Network Ingress Filtering proposal described in [14], traffic originating from a network is forwarded only if the source IP in the packets belongs to the network. Ingress filtering basically prevents a specific network from being used to attack others. Thus, while there is a collective social benefit in everyone deploying it, individuals do not receive direct incentives.

3. PROPOSED ALGORITHMS

A. BORDER GATEWAY PROTOCOL AND AS INTERCONNECTIONS:

BGP pis known for performing inter-domain routing in Transmission Control Protocol/Internet Protocol (TCP/IP) networks. BGP is an exterior gateway protocol (EGP), which means that it performs routing between multiple ASes or domains and exchanges routing and reachability information with other BGP systems. As BGP is an incremental protocol, it generates updates only in response to network events. In the absence of any event, no route updates are triggered or exchanged between neighbors, and the routing system is said to be in a stable state. Formally, Definition 1 (Stable Routing State): A routing system is in a stable state if all the nodes have selected a best route to reach other nodes and no route updates are generated (and propagated) by any node.

The Border Gateway Protocol (BGP) is one of the core routing protocol of the Internet. It maintains a table of IP networks or 'prefixes' which designate network reach ability among ASes. It is described as a path vector protocol.

To begin with, let us consider the AS graph of the Internet as an undirected graph G = (V, E). Each node v V corresponds to an AS (AS), and each edge e(u,v) E represents a BGP session between two neighboring ASes u,v V . It is assumed that there is at most one edge between neighboring ASes. BGP route updates which can be a announcement or withdrawal, are exchanged by nodes to learn the changes (if any) to reach destination ability prefixes. A route announcement list contains a list of route attributes associated with the destination network prefix. Of particular interest to us is the path vector attribute, as_path, which is the sequence of ASes that this route has been propagated over. Let r.as_path be used to denote the AS path attribute of route r and r.prefix the destination network prefix of r. Let r.as_path = (vk, vk-1…v1v0). The route was first announced (originated) by node v0, which owns the address space as described by r.prefix. Before arriving at node vk , the route was carried over nodes v1,v2,…vk-1 that order. Route r and its AS path r.as_path can be used interchangeably. For convenience, consider a specific destination node d, all route announcements and withdrawals are specific to the network prefixes owned by d. Notation d can also be used to denote the network prefixes owned by the node. Hence, a route r can be used to denote the route to reach the destination d.

1. Policies and Route Selection:

A single best route to the destination (if any) is selected and propagated by a node to its neighbors. BGP is a policy-based routing protocol, where both the selection and the propagation of best routes are guided by locally defined routing policies. Two distinct sets of routing policies are usually employed by a node: import policies and export policies. Neighbor-specific import policies are applied upon routes learned from neighbors, whereas neighbor-specific export policies are applied on locally-selected best routes before they are propagated to the neighbors. Let r be a route (to destination d) received at v from node u. The possibly modified route that has been transformed by the import policies can be denoted by import (v u) [{r}]. After the routes are passed through the import policies at node v, they are stored in v's routing table. The set of all such routes is denoted as candidateR (v,d): candidateR (v, d) = { r : import(v u)[{r}] {} r.prefix = d; u N(v)}. (1) Here, N(v) is the set of v's neighbors Out of all the members of candidateR (v,d), node v selects a single best route to reach the destination based on a defined procedure(see [3]). The outcome of the selection procedure at node v, i.e., the best route, is denoted as bestR (v,d), which reads best route to destination d at node v After selecting bestR (v,d) from candidateR(v,d), v exports the route to its neighbors after applying neighbor specific export policies. The export policies determine if a route should be forwarded to the neighbor, and if so, modify the route attributes according to the policies. (Section III-B) export(v u)[{r}]

denote the route sent to neighbor u by node v, after node v applies the export policies on route r.

Table -1: Import routing policies at an as

If (u1 ∈ customer(v) ∪ sibling(v)) and (u2 ∈ peer (v) ∪ provider(v)) then r1.local_pref>r2.local_pref

Table -2: Route export rules at an as

Export rules		r1	r2	r3	r4
Export routes to		Provider	Customer	Peer	Sibling
Learned from	Provider	No	Yes	No	Yes
	Customer	Yes	Yes	Yes	Yes
	Peer	No	Yes	No	Yes
	Sibling	Yes	Yes	Yes	Yes
Own routes		Yes	Yes	Yes	Yes

2. AS Relationships and Routing Policies:

A pair of ASs can enter into one of the following arrangements: [6], [8]

- Provider to customer: In this arrangement, provider AS gets paid by customer AS to carry its traffic. It is most common when the customer is smaller in size and provider is much larger in size than customer.

- Peer to peer: In a mutual peering agreement, the Ass decides to carry traffic from each other (and their customers).

- Mutual Peers: They do not carry transit traffic fo each other.

- Sibling to sibling: Two ASs can provide mutual transit service to each other. One sibling AS can be regarded as the provider of the other AS

In this paper it is assumed that each AS sets its import routing policies and export routing policies according to the rules specified in Table I [7] and Table II [6], [8], respectively. In Table I r1,r2 denote are the routes (to destination) received by node v from the neighbors u1,u2 respectively and customer(v), peer(v), provider(v) and sibling(v) denote the set of customers, peers, providers and siblings of node v respectively. According to Import routing policies from Table I, an AS will prefer the routes learned from customers or siblings over the routes learned from peers or providers.

In Table II, the columns marked with r1-r4 denote the export policies employed by an AS to announce routes to providers, customers, peers, and siblings, respectively. For instance, export rule r1 states that an AS will announce routes to its own networks, and routes learned from customers and siblings to a provider, but it will not announce routes learned from other providers and peers to the provider. Hence, number of possible paths between each pair of ASes is limited. The export policies described in Table I are not complete. In a few cases, less restrictive policies may be applied by ASes inorder to satisfy traffic engineering goals. For the moment, assume that all ASes follow the rules r1-r4 and that each AS accepts legal routes exported by neighbors. If AS b is a provider of AS a, and AS c is a provider of AS b, then c is an indirect provider of a, and a an indirect customer of c. Indirect siblings are defined in a similar fashion. Rules r1-r4 implies that an AS will distribute the routes to direct or indirect customers/siblings to its peers and providers. If e(u, v) bestR(s,d).as path, then u is the best upstream neighbor of node v for traffic from node s to destination d, and denote u as u = bestU(s, d, v). An edge from a provider to a customer AS is referred as a provider-to-customer edge, an edge from a customer to provider as a customer-to-provider edge, and an edge connecting sibling (peering) ASes as sibling-to-sibling (peer-to-peer) edge. A downhill path is a sequence of edges that are either provider-to-customer or sibling-to-sibling edges, and an uphill path is a sequence of edges that are either customer-to-provider or sibling-to-sibling edges. Gao [9] established the following theorem about the candidate routes in a BGP routing table. Theorem 1 (Gao [9]): If all ASes set their export policies according to r1-r4, any candidate route in a BGP routing table is either (a) an uphill path, (b) a downhill path, (c) an uphill path followed by a downhill path, (d) an uphill path followed by a peer-to-peer edge, (e) a peer-to-peer edge followed by a downhill path, or (f) an uphill path followed by a peer-to-peer edge, which is followed by a downhill path.

B. INTER DOMAIN PACKET FILTERS:

In this section, the idea behind the IDPF architecture, using BGP updates IDPF construction is described, and establishment of the correctness of IDPFs is discussed. Later the cases where ASes have routing policies that are less restrictive than ones in Tables I and II is also discussed. Let us assume that the routing system is in the stable routing state in this section.

Let M(s, d) denote a packet that has source address as s, and destination address as d. A packet filtering scheme decides if the packet can be forwarded or dropped based on certain criteria. One example is the route based packet filtering [4]:

Definition 2 (Route-Based Packet Filtering): Node v accepts packet M(s, d) forwarded from node u if and only if e(u, v) ϵ bestR(s, d). Else the source address of the packet is considered as spoofed, and the packet is discarded by v.

Even with the perfect routing information, the route based packet filter would not be able to identify spoofed packet [4], a valid packet filter should focus on not dropping any

valid packets and possessing a capability to limit spoofed packets. Accordingly correctness of packet filter is defined as follows.

Definition 3(Correctness of Packet Filtering): A packet filter is correct if it do not discard packets having valid source addresses when the routing system is stable.

Since valid packets from source 's' to destination 'd' will only traverse the edges on bestR(s, d), clearly the route based packet filtering is correct. Computation of route-based packet filters requires the knowledge of bestR(s, d) on every node, which is impossible in BGP. IDPF overcomes this problem.

1. IDPF Overview: A topological route between nodes s and d is a loop free path between the two nodes. Such routes (topological) are implied by network connectivity.

A topological route is a feasible route under BGP if and only if the construction of the route does not violate the routing policies imposed by the relationship between ASes (Table I and II). Let feasibleR(s, d) be the set of feasible routes from source s to destination d, it can be defined as follows:

$$\textbf{feasibleR(s, d)}) = \{ < s \oplus U \text{ feasibleR}(u, d) > \},$$

$$u: \text{import}(s \leftarrow u)[\{r\}] \neq \{\},$$

$$r.\text{prefix} = d, u \in N(s)$$

Where $\oplus$ is the concatenation operation, for example:

{ s $\oplus$ {<ab>, <uv>}} = {<sab>,<suv> }. It is noticed that feasibleR(s, d) contains all the routes between the pair that does not violate the import and export routing policies given in Table I and II. Obviously, bestR(s, d) $\in$ candidateR(s, d) feasibleR(s, d). Each of the feasible routes can potentially be a candidate route in a BGP routing table. Theorem 1 also applies to feasible routes.

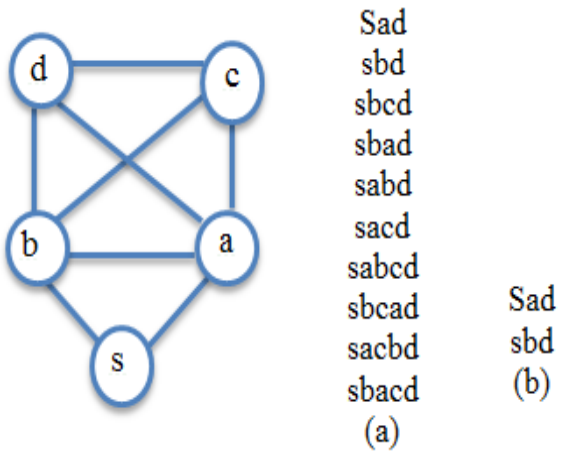

Fig -2: An example of network topology

(a)Topological routes implied by connectivity

(b) Feasible routes constrained by routing policies

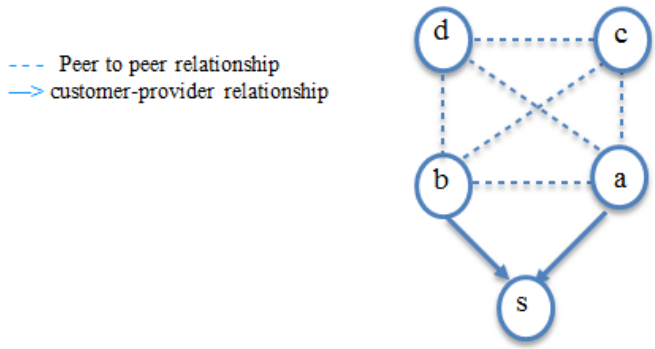

Fig 3. Routes between source s and destination d

Definition 4 (Feasible Upstream Neighbor): Consider a feasible route r ϵ feasibleR(s, d). If an edge e(u, v) ϵ r.as_path, then node u is a feasible upstream neighbor of node v for packet M(s, d). The set of all such feasible upstream neighbors of v is denoted as feasibleU(s, d, v).

The idea behind IDPF framework is as following:

Firstly, a node v is able to infer its feasible upstream neighbors using BGP route updates. This technique is described in next sub-section. A node can only allow M(s, d) from its feasible upstream neighbor to pass and discard all other packets since, bestR(s, d) ϵ candidateR(s, d) feasibleR(s, d). Such a filtering will not discard legitimate packets. Secondly, although network connectivity may imply a large number of topological routes between a source and destination, commercial relationship between ASes and routing policies employed by ASes act to restrict the size of feasibleR(s, d). Consider an example [13] in Figure 2. Figure 3(a) and 3(b) represent the topological routes implied by network connectivity and feasible routes constrained by routing between source s and d. In figure 3(b) it is assumed that nodes a, b, c and d have mutual peering relationship, and that a and b are providers to s. Although there are 10 topological routes between source s and d, there are only 2 feasible routes that are supported by routing policies in Table (I and II). Of more important to IDPF is that, feasible routes constrained by routing policies help limit the set of such neighbors. For an example, let us consider the situation at node d. Given that only nodes 'a' and 'b' (but not c) are on the feasible routes from s to d can infer that all packets forwarded by node c are spoofed and must be discarded. Hence IDPF is less powerful than route-based packet filters [4] because IDPF filters are computed based on feasibleR(s, d) instead of bestR(s, d). However, feasible routes can be inferred from local BGP updates while best routes cannot.

2. Construction of IDPF:

The technique to identify the feasible upstream neighbors of node v for packet M(s, d) is summarized by the following lemma.

Lemma 1: Consider a feasible route r between source s and destination d. Let v ∈ r.as_path and u be the feasible upstream neighbor of node v along r. When the routing system is stable, export (u-> v)[{bestR(u, s)}] ≠{}, assuming that all ASes follow the import and export routing policies in Tables I and II and that each AS accepts legitimate routes exported by neighbors [13]. Lemma 1 states that if node u is a feasible upstream neighbor of node v for packet M(s, d), node u must have exported to node v its best route to reach the source s [13].

Proof: Since Theorem 1 applies to feasible routes, a feasible route can be one of the six types of paths in Theorem 1. In the following it is assumed that the feasible route r is of type (c), i.e., an uphill path followed by a downhill path. Cases where r has other types (a), (b), (d)-(f) can be similarly proved. To prove the lemma, consider the possible positions of nodes u and v in the feasible route.

Case 1: Nodes u and v belong to the uphill path. Then node s must be an (indirect) customer or sibling of node u. From the import routing policies in Table I and the export routing policy r1 and the definition of indirect customers-siblings, it is followed that u will propagate to (provider) node v the reachability information of s [13].

Case 2: Nodes u and v belong to the downhill path. Let e(x, y) be the peer-to-peer edge along the feasible route r and note that u is an (indirect) customer of y. The node y learns the reachability information of s from x, based on the import routing policies in Table 1 and the export routing policy r3. From the export routing policy r2 and the definition of indirect customers, node y will propagate the reachability information of s to node u, which will further export the reachability information of s to (customer) node v [13].

Based on Lemma 1, a node can classify the feasible upstream neighbors for packet M(s, d) and consider the behavior of IDPF as follows:

Definition 5 (Inter-Domain Packet Filtering (IDPF)): Node v will accept packet M(s, d) forwarded by a neighbor node u, if and only if export (u -> v)[{bestR(u, s)}] ≠ {} [13]. Otherwise, the source address of the packet must have been spoofed, and the packet should be discarded by node v [13].

3. Correctness of IDPF:

Theorem 3: An IDPF as defined in Definition 5 is correct [13].

Proof: Without loss of simplification, consider source s, destination d, and a node v ∈ bestR(s, d).as_path such that v deploys an IDPF filter. In order to prove the correctness of the theorem, it is need to establish that v will not discard packet M(s, d) forwarded by the best upstream neighbor u, along bestR(s, d).

From the best route selection process, the best route between a source and destination is also a feasible route between the two (bestR(s, d) ∈ candidateR(s, d)) feasibleR(s, d)). Therefore, u is also a feasible upstream neighbor of node v for packet M(s, d). From Lemma 1, it is concluded that, export (u -> v) [{bestR(u, s)}] ≠{}, that is u must have exported to node v its best route to source s.

From Definition 5, packet M(s, d) forwarded by node u will not be discarded by v, and the correctness of the theorem is established [13].

4. Routing Dynamics:

Consider two different types of routing dynamics: 1) those caused by network failures; 2) and those caused by the creation of a new network (or recovery from a fail-down network event) [13].

For the first type of routing dynamics, the possibility that the filter will block a valid IP packet can be eliminated. It is illustrated as follows: Consider an IDPF facilitated AS v that is on the best route from s to d. Let u = bestU(s, d, v), and let U = feasibleU(s, d, v). If a link or router fails between u and s, it can have following three outcomes:

1) AS u can still reach AS s, and u is still preferred to be the best upstream neighbor for packet M(s, d), i.e, u = bestU(s, d, v). In this situation, although u may travel and announce multiple routes to v during the path exploration process [14], the filtering function of v is unaffected.

2) AS u is no longer the best upstream neighbor for packet M(s, d); another feasible upstream neighbor u' ∈ U can reach AS s and is instead chosen to be the new best upstream neighbor (for M(s, d)) [13]. Now, both u and u' may explore multiple routes; however, since u' has already announced a route (about s) to v, the IDPF at v can correctly filter (i.e., accept) packet M(s, d) forwarded from u' [13].

3) No feasible upstream neighbors can reach s. Therefore, AS v will also not be able to reach s, and v will no longer be on the best route between s and d. Hence, no new packet M(s, d) should be sent through v.

The second type of routing dynamics relates to how newly connected network or a network recovered from a fail-down event will be affected. In general, a network may start sending data immediately following the announcement of a new prefix, even before the route has time to propagate to the rest of the Internet [13]. In the time that it takes for the route to be propagated, some packets (from this prefix) may be discarded by some IDPFs if the reachability information has not yet propagated to them. In general, the time taken for new prefix information to reach an IDPF is proportional to the shortest AS path between the IDPF and the initiator of the prefix and independent of the number of alternate paths between the two [13]. Sometimes in the short timescales, it is acceptable for IDPFs to potentially behave incorrectly,

i.e. discarding valid packets originated from the new network prefix, before the corresponding BGP announcements reach the IDPFs.

But in this case, the packets are stored, till the packet is alive. Once the packet is time out, BGP announcements are sent to the IDPF's and the new feasible routes are found. The packet is then sent through those new feasible routes. Thus, the packet is not discarded by handling the network dynamically.

C. IP TRACEBACK

IP traceback is to find the origin of an IP packet on the Internet without trusting on the source IP address field. The source IP address of a packet is not authenticated, due to the trusting nature of the IP protocol. Therefore, the source address in an IP packet can be falsified (IP address spoofing). Spoofed IP packets can be used for different attacks. IP traceback problem is known as the problem of finding source of a packet. IP traceback is a critical ability for identifying sources of attacks and instituting protection measures for the Internet. Most existing approaches to this problem have been tailored toward DDoS attack detection.

Inter Domain Packet Filter is used to find the feasible routes. If the packet is not among the set of the feasible routes, it is detected as spoofed packet and is discarded. The spoofed packet's route is traced to detect the intruder

4. CONCLUSION AND FUTURE WORK

In this paper we proposed an inter-domain packet filter (IDPF) architecture as an effective guard against the IP Spoofing-based DDoS attacks. IDPFs rely on BGP update messages exchanged on the Internet to assess the validity of source address of a packet forwarded by a neighbor. The conditions under which IDPF framework can work correctly without discarding any valid packets are stated. It is observed that by deploying IDPFs on the Internet, the spoofing capability of attackers can significantly be limited. In addition, they also help to determine the true origin of an attack packet thus simplifying the IP traceback process.

REFERENCES

1. ICSNN/SSAC. "ICANN SSAC Advisory SAC008 DNS Distributed Denial of Service (DDoS) Attacks." Mar. 2006.
2. R. Beverly and S. Bauer, "The Spoofer Project: Inferring the extent of Internet source filtering on the internet," in Proceedings of Usenix SRUTI, Cambridge, MA, Jul. 2005.
3. J. Stewart, "DNS cache poisoning-the next generation," LURHQ , Techinical Report, Jan. 2003.
4. K. Park and H. Lee, "On the effectiveness of route-basedpacket filtering for distributed DoS attack prevention in power-law internets," in Proc.ACM SIGCOMM, San Diego, CA, Aug. 2001.
5. Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC 1771, Mar. 1995.
6. L. Gao, "On inferring autonomous system relationships in the internet," .IEEE/ACM Transactions on Networking, vol. 9, no. 6, Dec. 2001.
7. L. Gao and J. Rexford, "Stable internet routing without global coordination," IEEE/ACM Transactions on Networking, vol. 9, no. 6, Dec. 2001.
8. G. Huston, .Interconnection, peering and settlements-part I,"The Internet Protocol Journal, Mar. 1999.
9. F. Baker, "Requirements for ip version 4 routers," RFC 1812, Jun. 1995.
10. C. Jin, H. Wang, and K. Shin, "Hop-count filtering: an effective defense against spoofed ddos traffic," in Proceedings of the 10th ACM conference on Computer and communications security, Oct. 2003.
11. A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," in IEEE Symposium on Security and Privacy, May 2003.
12. StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense," IEEE Journal on Selected Areas in Communications, vol. 24, no. 10, Oct. 2006.
13. Zhenhai Duan, Member, IEEE, Xin Yuan, Member, IEEE, and Jaideep Chandrashekar, Member, IEEE, "Controlling IP Spoofing Through InterDomain Packet Filters",2009.
14. J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky, .Limiting path exploration in BGP,. in Proc. IEEE INFOCOM, Miami, FL, Mar. 2005.
15. Cisco Systems, Inc., "Unicast reverse path forwarding loose mode," http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newf% t/122t/122t13/ft urpf.pdf.